UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/733,788	12/10/2003	Mehmet Yunt	MWS-106RCE	8974

74321 7590 06/04/2008

LAHIVE & COCKFIELD, LLP/THE MATHWORKS
One Post Office Square
Boston, MA 02109-2127

EXAMINER	
SILVER, DAVID	

ART UNIT	PAPER NUMBER
2128	

MAIL DATE	DELIVERY MODE
06/04/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/733,788	YUNT ET AL.	
	Examiner	Art Unit	
	DAVID SILVER	2128	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 February 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-71* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-71* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-71 are currently pending in Instant Application.

2. The Instant Application is not currently in condition for allowance.

Priority

3. Priority is not claimed **(Effective Filing: 12/10/2003)**.

Response to Arguments

Response: 35 U.S.C. § 112

4. Applicants' amendments and arguments **(Remarks: page 19 bottom - page 20 top)** are sufficient to overcome the 35 U.S.C. § 112 rejection. Accordingly, the rejection is withdrawn.

Response: 35 U.S.C. § 103

5. **Applicants argue:**

5.1 "The debug information given in MathWorks relates to the blocks themselves and the state of the entire system - not the execution methods within blocks. Thus, MathWorks is focused a different level of granularity than claim 1. MathWorks does not disclose or suggest **said debug information indicating the order of execution of said plurality of execution methods for said block**, as present in claim 1. The amendment to claim 1 specifies that the execution methods are for a block. Further, what is claimed includes the display of the block' s execution methods, a feature not disclosed by MathWorks.

5.2 The Examiner states that "MathWorks implies but does not make explicit that one or more blocks includes a plurality of execution methods." The combination proposed by the Examiner still does not result in the method of claim 1. Even if MathWorks disclosed that one or more blocks includes a plurality of execution methods (Applicants respectfully disagree with the Examiner's statement), it still is not obvious to **present the order of execution methods**, as present in claim 1.

5.3 Still further, MathWorks fails to disclose or **suggest a start time or a stop time of said plurality of execution methods**, as present in claim 1. In fact, MathWorks is silent as to such a start time or a stop time.

5.4 MathWorks does not disclose or suggest yet other features of claim 1. For example, MathWorks does not disclose or suggest **said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of execution methods that are executed in said block during the execution of said model**. The Examiner asserts that MathWorks discloses this feature, but does not provide any support for that assertion (Office Action at 4). Applicants have nevertheless amended claim 1 to further clarify that the execution methods are executed **in said block**. As noted above, MathWorks is focused on blocks, and not execution methods within a block. Therefore, MathWorks is silent about debug information regarding the operation of **at least a subset of said plurality of execution methods**.

5.5 For at least the reasons given above, MathWorks does not disclose or suggest each and every element of independent claim 1. Claims 2-3, 5-9, 17-22, and 24 depend from claim 1 and, as such, include each and every element of claim 1. Thus, MathWorks does not disclose or suggest each and every element of claims 2-3, 5-9, 17-22, and 24." (Remarks: page 21)

6. **Examiner Response:**

6.1 Regarding subsection 1 *supra*, Official Notice was taken with respect to the limitation of blocks having plurality of execution methods. The Official Notice was not properly traversed, and is therefore, taken as admitted prior-art in accordance with the MPEP. See below for detailed explanation. Given that MathWorks discloses the order of execution of blocks, and the blocks contain plurality of execution methods, the order of the execution of methods is inherently provided. Take, for example, BLOCK1=E1,E2,E4,E5; BLOCK2=E6,E7,E9. From this, it is clearly seen that E1,E2,E4,and E5 are executed before E6,E7,E9. Therefore, an order of method execution has been established.

6.2 Regarding subsection 2 *supra*, Official Notice was taken with respect to "one or more blocks includes a plurality of execution methods". Applicants have presented, at best, a general allegation as to why this limitation is not obvious ("Applicants respectfully disagree"). This is not sufficient to traverse an Official Notice. Therefore, in accordance with MPEP 2144.03.C, the content of the official notice is taken to be admitted prior-art because Applicants traversal was inadequate. As for the obviousness

of the order of execution methods, this arguments have been addressed *supra* and are respectfully traversed as such.

6.3 Regarding subsection 3 *supra*, in 12-5 MathWorks states that one block can be stepped at a time, when stepping through the blocks, the simulation outputs the Tm (time). Just as in the example above where BLOCK1=E1,E2,E4,E5; BLOCK2=E6,E7,E9, stepping through BLOCK1 will result in the start time of E1, and stop time of E5. Therefore, the limitation "a start time or a stop time of said plurality of execution methods" is taught by MathWorks.

6.4 Arguments regarding subsection 4 *supra* have been fully considered but are unpersuasive. Specifically, no functionality / structure is necessitated by the limitation of "allowing the user to". See MPEP 2111.04.

7. Arguments regarding claims 25-27, 29-33, 41-46, and 48-69 are likewise unpersuasive as they are on the basis of what was addressed above.

8. **Applicants argue:**

8.1 "In the MathWorks debugger, a user can display a model's block execution order (MathWorks at page 12-16), but they cannot identify when an execution method is operating. In the MathWorks debugger, users can set breakpoints at the beginning or end of a block, thus allowing them to identify when an operation is performed by a block (MathWorks at page 12-9), but users cannot identify an operation performed by an execution method at a determined location in the execution method. In MathWorks, users can step through the simulation block- by-block (MathWorks at page 12-5) and thus **identify block-level errors**; however, MathWorks does not allow a user to identify an error related to the first execution method or the second execution method during execution of the computer-based model. In fact, MathWorks is silent as to these features. Thus, MathWorks does not disclose or suggest all the elements of claim 48.

8.2 Claims 49-53 depend from claim 48, and thus include each and every element of claim 48. In light of

the above arguments, Applicants respectfully request that the 35 U.S.C. § 103(a) rejection of claims 48-53 be withdrawn." (Remarks: page 24)

9. **Examiner Response:**

9.1 Applicants' arguments are respectfully traversed as follows. The methods are related to the blocks associated therewith (see rejection of Claim 1). Therefore, the identification of errors within a block identifies errors that are related to the methods associated therewith, which in this case are the execution methods. Accordingly, the arguments are respectfully traversed and the rejection is maintained.

10. Arguments regarding claim 54 are likewise unpersuasive as they are on the basis of what was addressed above.

11. **Applicants argue:**

11.1 "MathWorks discusses blocks and not root methods. Thus, MathWorks does not give a debugging result with visual identifiers related to the operation of root or child methods, but instead gives visual identifiers related to the execution of blocks (MathWorks at page 12-16). MathWorks does not disclose or suggest displaying status information related to root or child methods, but can display status information related to blocks and overall system states (MathWorks at pages 12-12, 12-14, and 12-16)." (Remarks: page 26)

11.2 "Instead of displaying a hierarchy of root and child methods, MathWorks displays a block execution order. For instance, compare MathWorks at page 12-16, showing the hierarchy of MathWorks, with Figure 18A of the Application, which depicts the parent-child hierarchy. The block execution order of MathWorks is silent as to with root or child methods." (Remarks: page 27)

11.3 "MathWorks does not display information about execution methods. MathWorks obtains information only on individual block states and overall system states (MathWorks at pages 12-12 to 12-14). Thus, MathWorks does not identify execution methods for a graphical icon, as present in claim 61. MathWorks is also silent as to displaying information about execution methods in a plurality

of regions, as present in claim 61." (Remarks: page 28)

12. **Examiner Response:**

12.1 Regarding subsections 1-3 *supra*, the reference-disclosed blocks are made of methods. Further, as stated by the Applicants, blocks are merely a group of methods (higher level of abstraction), and they are therefore functionally equivalent. Therefore, because the reference discloses the root-child (hierarchic nature) at a block level, it inherently does so at the level that is beneath it, the method-level. Accordingly, the arguments are traversed and the rejection is maintained.

13. **Applicants argue:**

13.1 "GNU gprof is a profiler used to determine which parts of a program are taking the most execution time (GNU gprof at 1). GNU gprof is used to profile programs, not models including blocks. Thus, GNU gprof does not disclose or suggest a debugger interfaced with a model view of a model being executed, said model comprising a block, nor does GNU gprof provide debug information indicating the order of execution of execution methods for blocks. Because GNU gprof does not disclose execution methods in blocks, a feature also missing from MathWorks, GNU gprof does not remedy the shortcomings of MathWorks with respect to at least the above- mentioned features of claim 1." (Remarks: page 29)

14. **Examiner Response:**

14.1 Applicants argue that gprof does not remedy the alleged shortcomings of claim 1. It is noted that gprof was applied to claims 4, 10-16, 23, 28, 34-40, and 47. The reference was not applied to claim 1. As traversed above, claim 1, as claimed, has no shortcomings and is anticipated as set-forth in the Office Action. Nevertheless, gprof indeed disclsoes a model comprising blocks having a plurality of execution methods. See, for example, page 16 section titled "The Annotated Source Listing" which disclsoe s"Compiling with ... augments your program with basic-block counting code, in addition to function counting code.". Further, attention is drawn to the section of "init_block" which shows the different execution methods of the block above it (deflate, ct_init, etc). Therefore, the

argument is respectfully traversed and the rejection is maintained.

Claim Interpretation

15. Limitations drawn to allowing, enabling or making optional a function's performance does not further limit a claim. As such, any prior art not explicitly prohibiting the performance of the function inherently anticipates the limitation. See MPEP 2111.04

Claim Objections

16. Claim 71 is objected-to for failing to meet USPTO claim formulation guidelines: missing trailing period (".").

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:
The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

The following is a quotation of the second paragraph of 35 U.S.C. 112:
The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

17. Claims 70-71 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The Specification does not disclose the claimed features. Particularly, the Specification does not adequately disclose that the debug information during execution of the model. Applicants have not, in the Remarks dated 2/15/08 specifically pointed out where such support exists in the disclosure. No such support was found; accordingly, the rejection is made herein as **necessitated by amendment**.

18. Claim 10 is recites the limitation " another block ". Because the plurality of the blocks was removed from the parent claim, the "another block" limitation now lacks antecedent basis. This rejection is necessitated by amendment. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

19. Claims 1-3, 5-9, 17-22, 24, 25-27, 29-33, and 41-46, 48-71 are rejected under 35 U.S.C. 103(a) as being unpatentable over MathWorks' Simulink "Dynamic System Simulation for MATLAB" "Using Simulink Version 2.2", 1997 ("MathWorks"), and further in view of Official Notice taken (admitted prior art).

As per claim 1, MathWorks discloses: In a modeling and execution environment, a method comprising the steps of:

providing a graphical debugger interfaced with a model view of a model being executed, said model comprising a block having at least a method **(12-3; 9-37; 9-42 "The block can integrate using these methods: ...")**, said graphical debugger having debug information related to the execution of said model **(12-3)**, said debug information indicating an order of execution of said plurality of execution methods for said block **(12-16, 12-16 to 12-19, 12-5)** and a start time or a stop time of said plurality of execution methods for said block that are executed in said block during the execution of said model **(start time ... 12-3 last para; 2-12; stop time ... 4-2 "An important advantage is that you can perform certain operations interactively while a simulation is running: You can modify many simulation parameters, including the stop time, the solver, and the maximum step size.")**; and outputting said debug information to a user, said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of execution methods that are executed during the execution of said model.

MathWorks implies but does not make explicit that "one or more blocks includes a plurality of execution methods" **(12-3; 9-37; 9-42 "The block can integrate using these methods: ...")**.

Official Notice is taken with respect to this feature - Office Action is admitted prior art due to inadequate traversal.

The legal basis for the 35 U.S.C. § 103 rejection is detailed in MPEP 2144.04.VI.B titled "Duplication of Parts", wherein it is described that mere duplication of parts has no patentable significance unless a new and unexpected result is produced. In this instance, merely duplicating the number of methods that each block contains does not produce a new and unexpected result.

Motivation to do so would have been to create a more compact design, which is also not a patentably significant feature. See MPEP 2144.04.V.B.

As per claim 2, Mathwork discloses: The method of claim 1, comprising the further steps of:

wrapping data generated by the execution of said model in an object, said wrapping encapsulating said execution-generated data in said object **(11-3: How to Specify a Path for a Simulink Object, 9-4 "To File", 9-61, -144, -145)**; and

exposing said data to said debugger via at least one interface to said object **(9-92 the exposure occurs when the debugger reads the information into the memory "From File", 9-61, -144, -145)**.

As per claim 3, MathWorks discloses: The method of claim 2, comprising the further step of: altering said data via said interface **(-131, 4-2: "An important advantage is that...")**.

As per claim 5, MathWorks discloses: The method of claim 1, comprising the further steps of:

processing said model to create compiled model information **(1-10 bullet 2, 1-12, 8-2: "C language S-functions are compiled as MEX-files using the mex utility described in the Application Program Interface Guide. As with other MEX-files, they are dynamically linked into MATLAB when needed.)**; and

programmatically generating executable code from said compiled model information, said code including an interface to said debugger **(1-12: linked, 8-36 first 3 para, 8-42: cg_sfun.h is**

included if the file is being used in conjunction with the Simulink Real-Time Workshop to produce a stand-alone or real-time executable.).

As per claim 7, MathWorks discloses: The method of claim 6, comprising the further steps of:

saving an execution history for said executable code **(MathWorks' "Target Language Compiler Reference Guide" ("TLC") futher expands on this inherent feature on page A-20 "This history is saved in the real-work vector.")**; and

outputting the execution history by at least one of saving it in a permanent memory location **(this feature is inherent)**, displaying it for a user **(the GUI displays the results to the users, furthermore, the data stored to the files is viewable by users)**, or sending it to a printing device to be printed **(RTW: 4-9, MathWorks: 3-26)**.

As per claim 8, MathWorks discloses: The method of claim 6 wherein said debugger is started after compilation and before the execution of said code **(this feature is inherent within the disclosure. Specifically, the debugger must have something to debug and therefore debugs after the compilation has finished. Furthermore, the debugger starts the execution of the code and is therefore started before the execution of the code.)**.

As per claim 9, MathWorks discloses: The method of claim 1, comprising the further step of:

indicating graphically using said debugger a plurality of blocks that are part of an algebraic loop when the executing model is processing the algebraic loop **(7-10, 12-14, 12-18, 4-20 first para)**.

As per claim 17, MathWorks discloses: The method of claim 1, comprising the further step of:

communicating with an external mode simulation with said debugger **(8-114: "SS_SIMMODE_EXTERNAL — External mode simulation")**.

As per claim 18, MathWorks discloses: The method of claim I, comprising the further step of:

saving a snapshot of data relating to model execution during execution of said model, said snapshot data sufficient to enable the subsequent restarting of the execution of said model using said snapshot data **(4-16: "You can also save the final states for a simulation and apply**

them to another simulation. This feature might be useful when you want to save a steady-state solution and restart the simulation at that known state.").

As per claim 19, MathWorks discloses: The method of claim 18 wherein said snapshot data is saved programmatically at least one or more of a regular interval or based on a user-defined parameter **(4-16: "You can also save the final states for a simulation and apply them to another simulation. This feature might be useful when you want to save a steady-state solution and restart the simulation at that known state." The user defined parameter is whenever the user chooses to do so manually.)**.

As per claim 20, MathWorks discloses: The method of claim 19, comprising the further step of: loading a saved snapshot into said debugger; and

executing a saved model based on said saved snapshot, said saved model executing from a point in time said snapshot was saved using information from said saved snapshot **(4-16: "You can also save the final states for a simulation and apply them to another simulation. This feature might be useful when you want to save a steady-state solution and restart the simulation at that known state.")**.

As per claim 21, MathWorks discloses: The method of claim 18, comprising the further step of: displaying graphically to a user the saved snapshot data **(this feature is inherent when the snapshot is restarted)**.

As per claim 22, MathWorks discloses: The method of claim 21, comprising the further step of

displaying graphically to a user at least one additional set of snapshot data without restarting the execution of said model **(This feature is inherent, it is the filename of the snapshot.)**.

As per claim 24, MathWorks discloses: The method of claim 18, comprising the further step of:

saving a difference between a set of current model execution data and a saved snapshot **(this feature is inherent. Specifically, when the simulation is restarted from a snapshot point and later saved it will be saved with the difference incorporated within the new snapshot.)**.

As per claim(s) 25-27, 29-33, 41-46, note the rejection of claim(s) 1-3, 5-9, 17-22, 24 above. The Instant Claim(s) is/are functionally equivalent to the above-rejected claim(s) and is/are therefore rejected under same prior-art teachings.

As per claim 48, note the rejection of claims 1-2 above. The Instant Claim recites substantially same limitations as the above-rejected claims and therefore rejected under same prior-art teachings, but for: identifying a first execution method operating in a first environment of a modeling application that executes a model, where the first environment is one of a text-based environment, a time-based block diagram, a state based block diagram, or a data-flow diagram **(B-2 model file ... text-based environment)**;

identifying a second execution method operating in a second domain, where the second domain differs from the first domain **(1-6: GUI-based tools for designing simulating and analyzing systems)**.

As per claims 49-50, note the rejection of claim 1-2 above. The Instant Claims recite substantially same limitations as the above-rejected claim and therefore rejected under same prior-art teachings.

MathWorks discloses: 51. The method of claim 48, further comprising:

displaying a hierarchy containing information about the first execution method or the second execution method, the hierarchy allowing a user to identify relationships between the first execution method and the second execution method, the first execution method and another execution method, or the second execution method and the another execution method **(1-3: "You can view the system at a high-level, then double-click on blocks to go down through the levels to see increasing levels of model detail.")**.

MathWorks discloses: 53. The method of claim 48, further comprising: identifying the first execution method or the second execution method using a visual indicator to identify when the first execution method or the second execution method is executing **(12-5)**.

As per claim 52, note the rejection of claims 1-2, 51, 53 above. The Instant Claim recites substantially same limitations as the above-rejected claims and therefore rejected under same prior-art teachings.

As per claim 54, note the rejection of claim 1 above. The Instant Claim recites substantially same limitations as the above-rejected claim and therefore rejected under same prior-art teachings.

MathWorks disclsoes: 55. (previously presented) A method, comprising:

identifying a first root method comprising one or more child methods, the first root method related to a graphical modeling application; identifying a second root method related to the graphical modeling application **(4-22: y, y1..yn)**;

running the first root method and the second root method in a graphical debugger to obtain information about the operation of the first root method or the second root method; and displaying a debugging result to a destination, the debugging result comprising visual identifiers related to the operation of the first root method, the one or more child methods or the second root method, error information about the first root method, the one or more child methods or the second root method, an execution result for the first root method, the one or more child methods or the second root method, or status information related to the first root method, the one or more child methods or the second root method **(8-46, 8-111, 11-2; 3-51 (systems list); 4-25; 11-15; A-7)**.

As per claim 56, note the rejection of claim 55 above. The Instant Claim recites substantially same limitations as the above-rejected claim and therefore rejected under same prior-art teachings.

MathWorks discloses: 57. The method of claim 56, wherein the displaying an indicator further comprises:

displaying a first symbol when the status is related to the first root method; and displaying a second symbol when the status is related to the one or more child methods or the second root method **(3-19; 6-14; -118; 10-15; A-7)**.

MathWorks discloses: 58. The method of claim 56, wherein the displaying an indicator further comprises:

displaying a first color to represent a first status related to the first root method; and displaying a second color to represent a second status related to one of the one or more child methods or the second root method **(3-19; 6-14; -118; 10-15; A-7)**.

MathWorks discloses: 59. The method of claim 56, further comprising:

displaying the hierarchy in a first region related to one or more display devices; and displaying a

graphical diagram related to the first root method or the second root method in a second region related to the one or more display devices, the graphical diagram synchronized with information displayed in the first region **(3-19; 6-14; -118; 10-15; A-7)**.

As per claim 60-66, note the rejection of claims 50-51, 57-60 above. The Instant Claim recites substantially same limitations as the above-rejected claims and therefore rejected under same prior-art teachings.

MathWorks discloses: 66. The method of claim 65, wherein the first indicator or the second indicator are a color, a pointer, a symbol, a font, or a border **(A-7)**.

MathWorks discloses: 67. The method of claim 64, wherein the first display area comprises a window that displays information about the graphical icon or the graphical icon debugging information **(2-6; 2-7; 2-11; 3-49)**.

MathWorks discloses: 68. The method of claim 67, wherein the window comprises a visual indicator to connect the window to the graphical icon or to the graphical icon debugging information **(2-6; 2-7; 2-11; 3-49)**.

MathWorks discloses: 69. The method of claim 64, further comprising: displaying an execution list in the hierarchy, the execution list related to the root method or the one or more child methods **(3-49)**.

MathWorks discloses: 70. (new) The method of claim 1, wherein the model comprises a plurality of blocks having execution methods, and wherein the debug information indicates an order of execution of said execution methods of said plurality of blocks, during execution of the model.

MathWorks discloses: 71. (new) The medium of claim 25, wherein the model comprises a plurality of blocks having execution methods, and wherein the debug information indicates an order of execution of said execution methods of said plurality of blocks, during execution of the model

20. Claim 4, 10-16, 23, 28, 34-40, 47 rejected under 35 U.S.C. 103(a) as being unpatentable over MathWorks's Simulink, 1997 ("MathWorks") as applied to claim 1 above, and further in view of Fenlason's "GNU gprof" ("GNU gprof") (1998).

As per claim 4, MathWorks discloses all limitations of claim 1, and that the execution-generated data is at least one of state information **(4-16 "Loading and Saving States", -131, A-22: signal generators, etc, 8-65)**, block inputs, block outputs **(3-15, 8-46 "In general, block inputs and outputs are written", 9-80)**, solver data **(4-4, 4-6, 4-16)**, signal values for said model **(-119, 8-124)**. MathWorks however does not explicitly disclose profiling data. GNU gprof however discloses an analogous application profiling system having the said feature **(page 14, "The primary line of this entry describes the total time spent directly in the functions of the cycle.")**. It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to time the execution of a program and routines of the program in order to identify which portions of the program cause a bottleneck and resolve them.

As per claim 10, MathWorks discloses: The method of claim 1, comprising the further step of: saving a record of a unique execution method invocation, **(1-3: "After you define a model, you can simulate it, using a choice of integration methods, either from the Simulink menus or by entering commands in MATLAB's command window.")**. MathWorks however does not substantially disclose said execution unique execution method invocation comprising information related to the execution of one of said plurality of execution methods that belongs to said block or to another block in the model, a system, or a model instance in an execution list of called execution methods. GNU gprof however discloses an analogous application profiling system having the said feature **(page 11: Call Graph)**.

As per claim 11, MathWorks discloses all limitations of claim 10. MathWorks does not expressly disclose that the unique execution method invocation record comprises information about child records of a subset of said plurality of execution executed inside said unique execution method invocation record. GNU gprof however discloses the said features **(page 12 section titled "children")**. It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to time the execution of a program and routines of the program in order to identify which portions of the program cause a bottleneck and resolve them.

As per claim 12, MathWorks discloses all limitations of claim 11. MathWorks however does not expressly

disclose that a link is provided from said unique execution method invocation record to said child record. GNU gprof however discloses an analogous system having the said feature **(page 6 section titled "--file-ordering map_file": "The `--file-ordering' option causes gprof to print a suggested .o link line ordering for the program based on profiling data.")**. It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to time the execution of a program and routines of the program in order to identify which portions of the program cause a bottleneck and resolve them.

As per claim 13, MathWorks discloses all limitations of claim 10. MathWorks does not however expressly disclose that the said unique execution method invocation record comprises information regarding at least one parent record of one or more of the plurality of execution methods in which said unique execution method invocation is executed. GNU gprof however discloses an analogous system having the said feature **(page 11: Call Graph)**. It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to time the execution of a program and routines of the program in order to identify which portions of the program cause a bottleneck and resolve them.

As per claim 14, MathWorks discloses all limitations of claim 13. MathWorks however does not expressly disclose a link is provided from said unique execution method invocation record to said parent record. GNU gprof however discloses an analogous system having the said feature **(page 6 section titled "--file-ordering map_file": "The `--file-ordering' option causes gprof to print a suggested .o link line ordering for the program, page 11: Call Graph)**. It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to time the execution of a program and routines of the program in order to identify which portions of the program cause a bottleneck and resolve them.

As per claim 15, MathWorks discloses all limitations of claim 10. MathWorks however does not expressly disclose that the said unique execution method invocation record comprises data about a state of the method invocation. GNU gprof however discloses an analogous system having the said feature **(page**

11: Call Graph - called column).

As per claim 16, MathWorks discloses all limitations of claim 15. MathWorks however does not expressly disclose that the said state indicates the method invocation is at one of the states of entering, entered, exiting and exited **(page 11: Call Graph).**

As per claim 23, MathWorks discloses all limitations of claim 22. MathWorks however does not expressly disclose that the said set of snapshot data is displayed in order of decreasing time. This is merely a design choice. Microsoft Windows allows for sort of descending or ascending names, file types, sizes, creation and modification dates. This is done for faster searching and identification of the user-required information.

As per claim(s) 28, 34-40, and 47, note the rejection of claim(s) 4, 10-16, and 23 above. The Instant Claim(s) is/are functionally equivalent to the above-rejected claim(s) and is/are therefore rejected under same prior-art teachings.

Support for Amendments and Newly Added Claims

Applicants are respectfully requested, in the event of an amendment to claims or submission of new claims, that such claims and their limitations be directly mapped to the specification, which provides support for the subject matter. This will assist in expediting compact prosecution. MPEP 714.02 recites: "Applicant should also specifically point out the support for any amendments made to the disclosure. See MPEP § 2163.06. An amendment which does not comply with the provisions of 37 CFR 1.121(b), (c), (d), and (h) may be held not fully responsive. See MPEP § 714." **Amendments not pointing to specific support in the disclosure may be deemed as not complying with provisions of 37 C.F.R. 1.131(b), (c), (d), and (h) and therefore held not fully responsive.** Generic statements such as "Applicants believe no new matter has been introduced" may be deemed insufficient.

Conclusion

21. All claims are rejected.

22. The Instant Application is not currently in condition for allowance.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David Silver whose telephone number is (571) 272-8634. The examiner can normally be reached on Monday thru Friday, 10am to 6:30pm. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kamini Shah can be reached on 571-272-2279. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Kamini S Shah/

Supervisory Patent Examiner, Art Unit 2128

/ DS /
David Silver, Patent Examiner
Art Unit 2128